REMARKS

Claims 1, 7, 14, 15, and 18-22 are currently pending. Claims 13 and 24-26 have been cancelled without prejudice or disclaimer. Claims 1, 7, 14, 15, and 18-22 have been amended. The amendment of claims 1, 7, and 22 is supported by page 5, line 20, through page 8, line 25, box 120 of Figure 2, and elsewhere in the originally filed disclosure. Claims 27-30 have been added and are supported by page 6, line 25, through page 7, line 4, and page 9, lines 1-10, of the application as filed. It is respectfully submitted that no new matter has been added.

Telephone Interview of June 26 2008

Patent Examiner Jeffery Williams and Applicant's representative Walter Malinowski held a telephone interview on June 26 2008. In the Examiner's opinion the claimed subject matter of a public key obtained from a source and a hidden public key from a loader program were obvious and known; however, Applicant's representative pointed out that the cited prior art does not teach both a public key and a hidden public key. The Examiner tentatively thought that a security configuration file or a policy configuration file having a digital signature which file is updatable using the private key might make the independent claims allowable.

In a follow up telephone interview on June 26 2008, subject matter found on page 9, lines 1-10, in conjunction with page 6, line 25, through page 7, line 4, and box 120 of drawing Figure 2 was discussed by the Examiner and Applicant's Representative. The Examiner said that he could not comment on the patentability of this subject matter.

Objection to the Specification

The Patent Office objected to the specification as failing to provide antecedent basis for claimed subject matter found in claims 1, 7, 13-15, and 18-22; specifically, "wherein the verification system ... provides security integrity for the virtual machine it installs," "wherein the verification system verifies the authenticity of each element of a virtual machine...," and "wherein the virtual machine installation installs a Java Virtual Machine."

Although Applicant has further amended the above noted language, Applicant wishes to point out that page 9, lines 12-13, of the application discloses "verification of the authenticity of

a JVM." Applicant's invention is stated on page 3, lines 4-6, as alleviating disadvantages such as "security integrity," disclosed on page 1, line 26, and page 2, lines 11 and 27, of Applicant's application as filed. Further, the abstract recites as follows:

> A software security system is arranged to **verify the authenticity of each element of a Java Virtual Machine installation**. A digital signature is attached to each file of the JVM installation. A loader (20) verifies the digital signature of the JVM DLL (30). The JVM DLL 30 then verifies the digital signature of each other DLL and configuration file to be loaded (40, 50, 60, 70), and only loads those files which have successfully verified digital signatures. In this way the security of the JVM is enhanced, a user has greater confidence that the Java applications will function correctly, and the detection of incorrect or damaged JVM installations is improved.

Claim Rejections under 35 U.S.C. 112, first paragraph

The Patent Office rejected claims 1, 7, 13-15, 18-22, and 24-26 under 35 U.S.C. 112, first paragraph, allegedly as failing to comply with the written description requirement.

Since no particular language is described as needing support, Applicant presumes that the language in question is that discussed and answered in the objection to the specification above.

Claim Rejections under 35 U.S.C. 112, second paragraph

The Patent Office rejected claims 1, 7, 13-15, 18-22, and 24-26 under 35 U.S.C. 112, second paragraph, as being indefinite for failing to point out and distinctly claim the subject matter which applicant regards as the invention.

The expressions "verification system" and "verification method" are not nonsensical. Upon reflection, Applicant has revised the preambles to recite both verification and installation since these two activities are intertwined in Applicant's claimed and disclosed invention.

Regarding claims 24, 25, and 26, the language "virtual machine installation installs a Java Virtual Machine" is not nonsensical. Noting that the Patent Office is permitted to use the broadest reasonable interpretation of claim language, Applicant wishes to point out that Webster's College Dictionary, for instance, provides three meanings for "installation": 1) the act of installing : the state of being installed; 2) something that is installed; and 3) a military camp, fort, or base. It should be clear that "installation" most nearly means in claims 24, 25, and 26

"the act of installing."

Reference by DLL of next Level DLL using Digital Signature

The Patent Office rejected claims 1, 2, 5, 7, 8, 11, and 13-23 under 35 U.S.C. 103(a) as being unpatentable over Shear, U.S. Patent No. 6,157,721, in view of Bodrov, U.S. Patent No. 6,802,006.

The present claimed invention, as expressed by independent claims 1, 7 and 22, is directed to providing security for the installation of a Java Virtual Machine.

Reference by DLL of next Level DLL using Digital Signature

Neither Shear nor Bodrov disclose or suggest using a library file of one level to verify authenticity of a library file of another level.

Claim 1 recites in pertinent part as follows: "the primary library file subsequently verifying and selectively loading the plurality of secondary files by calling the loader program to compare the obtained digital signature key with the digital signature of each of the plurality of secondary files."

Claim 7 recites, in pertinent part, as follows:

> for each of a plurality of secondary files, using the primary library file to verify authenticity of a digital signature incorporated in corresponding one of the plurality of secondary files by calling the Java launcher to compare the digital signature incorporated in the corresponding one of the plurality of secondary files with the digital signature key; and, selectively loading the plurality of secondary files in dependence upon the successful verification of their digital signatures

Claim 22 recites in pertinent part as follows: "the virtual machine primary library file subsequently verifying and selectively loading the plurality of secondary files by calling the loader program to compare the obtained digital signature key with the digital signature of each of the plurality of secondary files."

Shear teaches a verifying authority 100 and a protected execution environment 108. The verifying authority 100 in Shear analyzes a load module 54 to ensure proper performance and upon passing tests assigns a seal of approval (or, digital signature) to the passed load module 54

(column 9, lines 42-66). In an embodiment where there are three segments 55(1), 55(2), and 55(3) of a load module, Shear discloses that the digital signatures of each segment may be verified by the same verifying authority 100 or three different verifying authorities (column 16, lines 12-36).

Shear discloses techniques for certifying load modules such as executable computer programs or fragments by a protected or secure processing environment (column 1, lines 25-28). Shear discloses (column 9, lines 42-51):

> FIG. 2 shows how a verifying authority 100 can prevent the problems shown in FIG. 1. In this example, authorized provider 52 submits load modules 54 to verifying authority 100. Verifying authority 100 carefully analyzes the load modules 54 (see 102), testing them to make sure they do what they are supposed to do and do not compromise or harm system 50. If a load module 54 passes the tests verifying authority 100 subjects it to, a verifying authority may affix a digital "seal of approval" (see 104) to the load module.

Shear further discloses (column 5, lines 10-25):

> A web of trust may stand behind a verifying authority. For example, a verifying authority may be an independent organization that can be trusted by all electronic value chain participants not to collaborate with any particular participant to the disadvantage of other participants. A given load module or other executable may be independently certified by any number of authorized verifying authority participants. If a load module or other executable is signed, for example, by five different verifying authority participants, a user will have (potentially) a higher likelihood of finding one that they trust. General commercial users may insist on several different certifiers, and government users, large corporations, and international trading partners may each have their own unique "web of trust" requirements. This "web of trust" prevents value chain participants from conspiring to defraud other value chain participants.

Shear does not teach or suggest that "the primary file subsequently verifying and selectively loading the plurality of secondary files by calling the loader program to compare the obtained digital signature key with the digital signature of each of the plurality of secondary files," as claimed in claim 1.

The Patent Office asserted on page 6, lines 9-11, of the February 27 2008 Office Action as follows: "Bodrov discloses that software modules, such as digitally signed DLLs defining a

software installation, interact via one module referencing another module to be loaded and verified by the loader program (Bodrov, fig. 2; 3:12-24)."

Bodrov shows a dynamic connection between two executable images 100, 200 (Figure 2). Bodrov, in Figure 1, shows an executable image 100 that is a data object and that is dynamically connectable with other executable images. Quoting from Bodrov (column 3, lines 12-24):

> The executable image 100 is a data object that can define by itself or in conjunction with other executable images, one or more software applications. The software applications may include, for example: a word processor, a database, a digital rights management system, a personal finance utility, a graphics tool, an Internet browser, a computer game, a communications program, an authorization program, an electronic wallet, a multi-media renderer or a contract manager. Furthermore, the executable image 100 is dynamically connectable with other executable images. For example, in an embodiment of the invention that is developed for use with the Windows 95, the executable image is a dynamic link library (DLL).

Although not stated, Applicant believes that this passage is meant to address the claimed subject matter of "a plurality of secondary files referenced by the primary library file, each of the plurality of secondary files having a digital signature." This claimed subject matter is part of independent claim 1, which also recites as follows:

> wherein the loader program verifies and selectively loads the primary library file by comparing the obtained digital signature key with the digital signature of the primary library file, the primary library file subsequently verifying and selectively loading the plurality of secondary files by calling the loader program to compare the obtained digital signature key with the digital signature of each of the plurality of secondary files

Just because Bodrov in Figure 2 shows executable image 100 pointing to executable image 200 does not mean that executable image 100 acts in the claimed manner of "the primary library file subsequently verifying and selectively loading the plurality of secondary files by calling the loader program..." Bodrov in column 3, lines 12-24, discloses that executable image 100 is dynamically connectable with other executable images, but does not teach or suggest "the primary library file subsequently verifying and selectively loading the plurality of secondary files by calling the loader program

Bodrov teaches a loader program 208 "to copy an executable image 100 from the storage device 105 (FIG. 1) to the memory 108 and to bind the code and data pointers to an appropriate address prior to the execution o f the executable image" (col. 3, lines 49-53). The validator 204 of Bodrov may be an executable image similar in format to the executable image 100, integrated with the executable image 100, or integrated with the program loader 208 (column 3, lines 43-49).

"The validator 204 analyzes the executable image 100 before the executable image 100 is loaded into the memory 108 and generates a reference digital signature with respect to the executable image 100" (column 4, lines 17-20). "After the executable image 100 is loaded, the validator 204 generates an authenticity digital signature to ensure that the executable image 100 has not been tampered with" (column 4, lines 21-23).

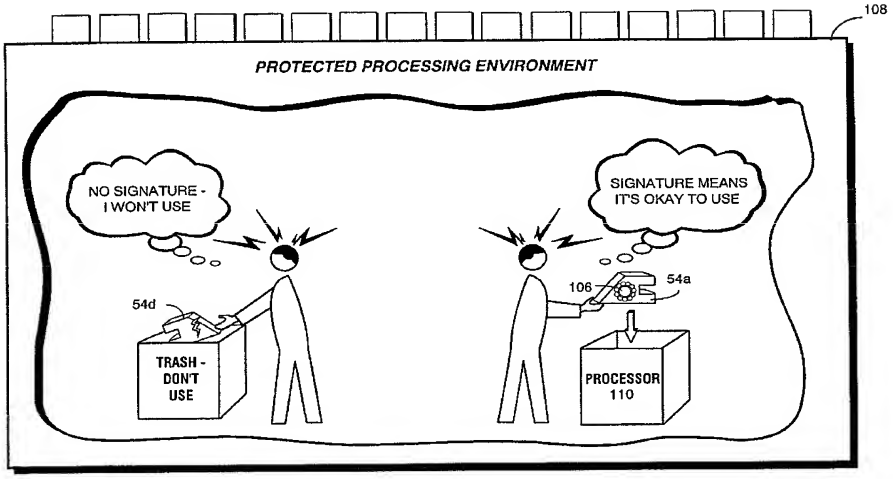
The Patent Office has asserted that Figure 3 and column 6, lines 5-15, of Shear and Figure 1 of Bodrov make teach or suggest the claimed subject matter where a primary library verifies and selectively loads a secondary library file by calling the loader program to compare the digital signature key with the digital signature of a secondary library file. Applicant does not understand how either or both drawing figures or the passage from Shear provides such a teaching or suggestion.

Shear does not teach or suggest the above claimed subject matter, but, rather, in column 6, lines 5-15, discloses as follows:

> In accordance with another aspect provided by the present invention, an execution environment protects itself by deciding--based on digital signatures, for example--which load modules or other executables it is willing to execute. A digital signature allows the execution environment to test both the authenticity and the integrity of the load module or other executables, as well permitting a user of such executables to determine their correctness with respect to their associated specifications or other description of their behavior, if such descriptions are included in the verification process.

Shear does not teach or suggest the above claimed subject matter, but, rather, in Figure 3 shows a single level step where a verified load module is used and an unsuccessfully verified load module is discarded. There is no suggestion or teaching of a primary library file verifying authenticity of a secondary library file. Shear's Figure 3 is reproduced immediately below:

FIG. 3 Before Protected Processing Environment Uses A Load Module, It Checks To See If Load Module Has Been Verified



The relevance of Bodrov's Figure 1 is not understood as to the noted claimed subject matter. Bodrov's Figure 1, reproduced immediately below, does not appear to disclose, show, or suggest a primary library file and a secondary library file which is verifiable by the primary library file:

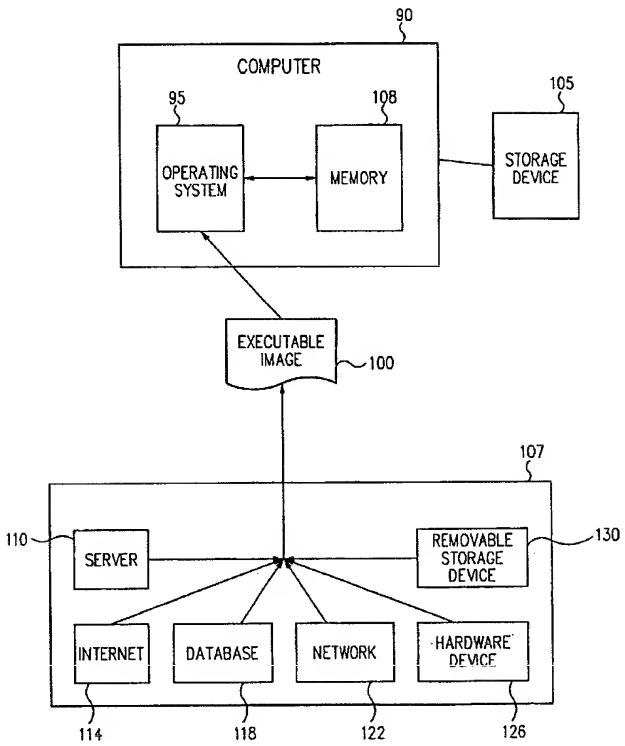


FIG. 1

Public Key and Hidden Public Key

It does not appear that Shear in column 13, line 65, through column 14, line 5, or in column 5, lines 1-5, teaches the claim language it is purported to teach. **Shear does not teach that the digital signature key may be one of two alternatives: a hidden public key internal to the loader program or the public key obtained via the virtual machine provider.** In Applicant's claimed invention, there are two potential sources of a digital signature key to determine the loading of the primary library file; in Shear, there is a first key to lock a message and a second key to unlock the message. In Shear, column 13, lines 30-49, the first key or private key locks the encrypted message and the second key or public key opens the locked encrypted

message. Figure 5 shows the private or first key 122 used to create a certifying digital signature; Figure 6 shows authenticating a digital signature from within a protected processing environment. As Shear discloses in column 13, on lines 54-56, "[o]nce message digest 116 is locked into strong box 118 using the first key 122 the strong box can be opened only by using the corresponding second key 124."

The Patent Office asserted as follows on page 3, lines 10-11, of the February 27 2008 Office Action: "(Shear, 13:65-14:5; 5:1-5; Herein, Shear discloses both obtaining a hidden public key and obtaining a certificate from the software provider.)"

Even if Shear (or, Bodrov) were to teach a public key and a hidden public key (it is not admitted that this is taught), neither reference nor the references in combination teach or suggest the claimed subject matter of "if a public key cannot be obtained via a virtual machine provider over the internet, the digital signature key is a hidden public key internal to the loader program and, if a public key can be obtained via the virtual machine provider, the digital signature key is the public key obtained via the virtual machine provider over the internet."

Request for Reference providing a Teaching

The Patent Office on page 7, lines 7-8, asserted that a referenced module may reference another module. Claim 1 recites "the primary library file subsequently verifying and selectively loading the plurality of secondary files by calling the loader program to compare the obtained digital signature key with the digital signature of each of the plurality of secondary files." This subject matter does not appear to be taught or suggested by Shear or Bodrov, alone or in combination. The Patent Office appears to be taking unofficial "Official Notice." Applicant requests a proper teaching for the referenced subject matter from claim 1 which is also found in the other two independent claims: 7 and 22.

Claimed Invention is Directed to Security Integrity of an Installed Java Virtual Machine

Applicant has identified problems with the current art; e.g., a hacker can alter the behavior of the JVM outside the JVM environment so as to undermine the whole Java security model (page 1, line 25, through page 2, line 4) such as by disabling the security code or by

inserting destructive routines into the core of the JVM (page 2, lines 13-17). The present invention provides a scheme for verification of the authenticity of a JVM using digital signatures and offers advantages. These advantages include 1) enhanced security of the JVM, 2) greater user confidence in the correct function of Java applications, and 3) improved detection of incorrect or damaged JVM installations (page 9, line 20, through page 10, line 2, of Applicant's specification).

Applicant has disclosed the following in the background of the invention:

> In the field of this invention it is known that `Java 2` includes a significantly enhanced security model, compared to previous Java Virtual Machines (JVMs). This new model can restrict the behaviour of a Java applet or application to a clearly defined set of safe actions. This allows a user to download application code from the internet or across a computer network, and run the code in a previously installed JVM. The user can be confident that the application will be assigned the required privileges to function correctly, but to neither damage the user's machine nor divulge sensitive information held on that machine to others via the network or internet. **However, a problem with this approach is that the JVM itself must retain its security integrity in order to ensure downloaded code is restricted in this way. If a malicious user (hacker) has been able to gain access to the user's machine outside of the JVM environment and alter the behaviour of the JVM the whole Java security model is undermined.** For example, the hacker could alter the privileges assigned for software code from a specific source, thereby allowing subsequently downloaded code from this source to function beyond the limits otherwise set by the JVM, and such enhanced privileges could easily be configured to compromise the security integrity of the user's machine. Similarly, the hacker could disable the security code altogether, or worst still insert destructive routines into the core of the JVM which could be activated by an external trigger, such as specific time/date, or when other (possibly harmless) code is being executed. It is clear that with this malicious activity, early detection of such a compromise of the JVM core would be very useful, and could prevent more serious subsequent damage. If a malicious user decides to attack a machine, the JVM is an obvious target due to its significance in relation to web-based applications, servers and the like. Therefore the security integrity of the JVM is a highly significant factor in the security of the computer as a whole. **A need therefore exists for a software verification system, method and computer program element wherein the abovementioned disadvantages may be alleviated.**

Neither Shear nor Bodrov, alone or in combination, address the problem that Applicant has provided a solution for.

Thus, it is respectfully submitted that claims 1, 2, 5, 7, 8, 13-15, 18-22, and 24-26 are allowable over Shear and Bodrov.

Claims 14 and 15

Claims 14 and 15 recite "the virtual machine provider is accessed through an internet site to provide the public key." Even if Shear could be considered to provide a certificate, Shear, in the abstract, Figure 1, column 2, lines 33-40; column 3, lines 10-15 and 21-35; and column 5, lines 3-5, or elsewhere, does not disclose or suggest the virtual machine provider provides a public key through an internet site. Bodrov does not appear to remedy this deficiency. Thus, claims 14 and 15 are allowable over the prior art for this additional reason.

Response to Response to Arguments on Pages 9-10 of February 27 2008 Office Action

(i) As noted above, Applicant's invention relates to security integrity in installing a Java Virtual Machine. Neither Shear nor Bodrov teach or suggest such.

(ii) As noted above, neither Shear nor Bodrov disclose or suggest using a library file of one level to verify authenticity of a library file of another level. That is, first, as discussed above, neither Shear nor Bodrov, alone or in combination, disclose or suggest, a primary library file used to verify authenticity of a secondary library file or a secondary library file used to verify the authenticity of a tertiary library file.

(iii) Claims 14 and 15 recite as follows: "the virtual machine provider is accessed through an internet site to provide the public key." Applicant had also written on page 12 as follows:

> Even if Shear could be considered to provide a certificate, **Shear**, in the abstract, Figure 1, column 2, lines 33-40; column 3, lines 10-15 and 21-35; and column 5, lines 3-5, or elsewhere, **does not disclose or suggest the virtual machine provider provides a public key through an internet site. Bodrov does not appear to remedy this deficiency.** Thus, claims 14 and 15 are allowable over the prior art for this additional reason.

MPEP 2143.03 states as follows: "All words in a claim must be considered in judging the patentability of that claim against the prior art."

The Patent Office is respectfully requested to reconsider and remove the rejections of the claims 1, 7, 13-15, 18-22, and 24-26 under 35 U.S.C. 103(a) based on Shear in view of Bodrov,

and to allow all of the pending claims 1, 7, 13-15, and 18-22 as now presented for examination. An early notification of the allowability of claims 1, 7, 13-15, and 18-22 is earnestly solicited.

Respectfully submitted:

Walter J. Malinowski June 27, 2008

Walter J. Malinowski Date

Reg. No.: 43,423

Customer No.: 29683

HARRINGTON & SMITH, PC
4 Research Drive
Shelton, CT 06484-6212

Telephone: (203) 925-9400, extension 19
Facsimile: (203) 944-0245
email: wmalinowski@hspatent.com